

IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD**Publication number:** JP2006515941 (T)**Publication date:** 2006-06-08**Inventor(s):****Applicant(s):****Classification:****- International:** G06F15/80; G06F9/30; G06F9/38; G06F9/445; G06F9/46;
G06F15/78; G06F15/76; G06F9/30; G06F9/38; G06F9/445;
G06F9/46**- European:** G06F9/38S4**Application number:** JP20050502222T 20031031**Priority number(s):** US20020422503P 20021031; US20030683929 20031009;
US20030683932 20031009; US20030684053 20031009;
US20030684057 20031009; US20030684102 20031009;
WO2003US34555 20031031**Also published as:** WO2004042660 (A2) WO2004042560 (A3) WO2004042574 (A2) WO2004042574 (A3) WO2004042569 (A2)[more >>](#)

Abstract not available for JP 2006515941 (T)

Abstract of corresponding document: WO 2004042560 (A2)

A peer-vector machine includes a host processor and a hardwired pipeline accelerator. The host processor executes a program, and, in response to the program, generates host data, and the pipeline accelerator generates pipeline data from the host data. Alternatively, the pipeline accelerator generates the pipeline data, and the host processor generates the host data from the pipeline data. Because the peer-vector machine includes both a processor and a pipeline accelerator, it can often process data more efficiently than a machine that includes only processors or only accelerators. For example, one can design the peer-vector machine so that the host processor performs decision-making and non-mathematically intensive operations and the accelerator performs non-decision-making and mathematically intensive operations. By shifting the mathematically intensive operations to the accelerator, the peer-vector machine often can, for a given clock frequency, process data at a speed that surpasses the speed at which a processor-only machine can process the data.

Data supplied from the esp@cenet database --- Worldwide

(19) 日本国特許庁(JP)

(2) 公 積 特 許 公 告(A)

(3) 特許出願公表番号

特表2006-515941

(P2006-515941A)

(4) 公表日 平成18年6月8日(2006.6.8)

(5) Int. Cl.

G06F 15/80 (2006.01)

F 1

G06F 15/80

テーマコード(参考)

審査請求 未請求 予備審査請求 未請求 (全 27 頁)

(21) 出願番号	特願2006-502222 (P2006-502222)	(71) 出願人	604275476 ロックヒード マーティン コーポレーション アメリカ合衆国 バージニア州 20110 マナサッカ、ゴッドウィン ドライブ 9500, メイル ドロップ 043, ビルディング 400
(30) (22) 出願日	平成15年10月31日 (2003.10.31)	(74) 代理人	100983932 弁理士 廣江 武典
(35) 離脱文書提出日	平成17年6月16日 (2005.6.16)	(74) 代理人	100129698 弁理士 武川 隆宣
(36) 國際出願番号	PCT/US2003/034555	(74) 代理人	100129676 弁理士 ▲高▼荒 新一
(37) 國際公開番号	W02004/042861	(74) 代理人	100120074 弁理士 中村 葵元
(38) 國際公開日	平成16年5月21日 (2004.5.21)		
(39) 優先権主張番号	60/422,503		
(40) 優先日	平成14年10月31日 (2002.10.31)		
(41) 優先権主張国	米国 (US)		
(42) 優先権主張番号	10/683,939		
(43) 優先日	平成15年10月9日 (2003.10.9)		
(44) 優先権主張国	米国 (US)		
(45) 優先権主張番号	10/683,932		
(46) 優先日	平成15年10月9日 (2003.10.9)		
(47) 優先権主張国	米国 (US)		

最終頁に続く

(64) 【説明の名称】多數バイブライン・ユニットを有するバイブルайн加速器、関連計算マシン、並びに、方法

(57) 【要約】

バイブルайн加速器は、バスと複数のバイブルайн・ユニットを含み、各ユニットがバスと結合されていると共に少なくとも1つの対応するハードウェアに組み込まれたバイブルайн回路を含む。複数のバイブルайн・ユニットをバイブルайн加速器に含ませることによって、単一のバイブルайн・ユニット加速器と比較して加速器のデータ処理性能を増大することができる。更には、複数のバイブルайн・ユニットが共通バスを介して通信するようにそれらバイブルайн・ユニットを設計することによって、バイブルайн・ユニットの数を変えることができ、よって、單にそれらバイブルайн・ユニットをバスと結合するか或は該バスから結合解除することによって加速器のコンフィギュレーション及び機能を変えることができる。これは、それらバイブルайн・ユニットの内の1つを変えるたび、或は、加速器内のバイブルайн・ユニットの数を変えるたびに、それらバイブルайн・ユニット・インターフェースを設計或は再設計する必要性をなくする。

【特許請求の範囲】**【請求項 1】**

パイプライン加速器であって、
通信バスと、

各々が、前記通信バスと結合されると共に各ハードウェアに組み込まれたパイプライン回路を含む複数のパイプライン・ユニットと、
を備えるパイプライン加速器。

【請求項 2】

前記パイプライン・ユニットの各々が、

前記ハードウェアに組み込まれたパイプライン回路と結合された各メモリを備え、 10

前記ハードウェアに組み込まれたパイプライン回路が、

前記通信バスからデータを受信し、

前記データを前記メモリにロードし、

前記メモリから前記データを検索し、

前記検索データを処理し、

前記処理データを前記通信バスに駆動するように動作可能である、

請求項 1 に記載のパイプライン加速器。

【請求項 3】

前記パイプライン・ユニットの各々が、

前記ハードウェアに組み込まれたパイプライン回路と結合された各メモリを備え、 20

前記ハードウェアに組み込まれたパイプライン回路が、

前記通信バスからデータを受信し、

前記データを処理し、

前記処理データを前記メモリにロードし、

前記処理データを前記メモリから検索し、

前記検索データを前記通信バスにロードすることができる、

請求項 1 に記載のパイプライン加速器。

【請求項 4】

前記ハードウェアに組み込まれたパイプライン回路の各々が各フィールド・プログラマブル・ゲート・アレイ上に配置されている、請求項 1 に記載のパイプライン加速器。 30

【請求項 5】

パイプライン・バスと、

前記通信バスと前記パイプライン・バスとに結合されたパイプライン・バス・インターフェースとを更に備える、請求項 1 に記載のパイプライン加速器。

【請求項 6】

前記通信バスが、各々が各パイプライン・ユニットと結合されている複数の分岐を含むことと、

前記分岐の各々と結合されたルータと、

を更に含む、請求項 1 に記載のパイプライン加速器。

【請求項 7】

前記通信バスが、各々が各パイプライン・ユニットと結合されている複数の分岐を含むことと、

前記分岐の各々と結合されたルータと、

パイプライン・バスと、

前記ルータと前記パイプライン・バスとに結合されたパイプライン・バス・インターフェースと、

を更に含む、請求項 1 に記載のパイプライン加速器。

【請求項 8】

前記通信バスが、各々が各パイプライン・ユニットと結合されている複数の分岐を含むことと、

40

50

前記分岐の各々と結合されたルータと、
パイプライン・バスと、

前記ルータと前記パイプライン・バスとに結合されたパイプライン・バス・インターフェースと、

前記ルータと結合された第2バスと、
を更に含む、請求項1に記載のパイプライン加速器。

【請求項9】

前記通信バスが前記パイプライン・ユニットの内の1つにアドレスされたデータを受信するように動作でき、

前記1つのパイプライン回路が前記データを受領するように動作でき、

10

前記他のパイプライン回路が前記データを拒絶するように動作できる、請求項1に記載のパイプライン加速器。

【請求項10】

前記通信バスが、各々が各パイプライン・ユニットと結合されている複数の分岐を含むことと、

前記分岐の各々と結合されたルータであり、

前記パイプライン・ユニットの内の1つにアドレスされたデータを受信し、

前記通信バスの前記各分岐を介して前記1つのパイプライン・ユニットに前記データを提供するように動作できるルータと、

を含む、請求項1に記載のパイプライン加速器。

20

【請求項11】

計算マシンであって、
プロセッサと、

前記プロセッサと結合されたパイプライン・バスと、
パイプライン加速器であり、

通信バスと、

前記パイプライン・バス及び前記通信バスの間に結合されたパイプライン・バス・インターフェースと、

各々が前記通信バスと結合されると共に各ハードウェアに組み込まれたパイプライン回路を備える複数のパイプライン・ユニットと、を含むことから成るパイプライン加速器と、

30

を備える計算マシン。

【請求項12】

前記プロセッサが前記パイプライン・ユニットの内の1つを識別するメッセージを生成して、該メッセージを前記パイプライン・バスに駆動するように動作でき、

前記パイプライン・バス・インターフェースが前記メッセージを前記通信バスに結合するように動作でき、

前記パイプライン・ユニットの各々が前記メッセージを分析するように動作でき、

前記識別されたパイプライン・ユニットが前記メッセージを受領するように動作でき、

前記他のパイプライン回路が前記メッセージを拒絶するように動作できる、請求項11に記載の計算マシン。

40

【請求項13】

前記通信バスが、各々が各パイプライン・ユニットと結合されている複数の分岐を備すことと、

前記プロセッサが前記パイプライン・ユニットの内の1つを識別するメッセージを生成して、該メッセージを前記パイプライン・バスに駆動するように動作できることと、

前記分岐の各々と結合されると共に前記パイプライン・バス・インターフェースと結合されて、前記パイプライン・バス・インターフェースから前記メッセージを受信してそのメッセージを前記識別されたパイプライン・ユニットに提供するように動作できるルータと、

50

を更に含む、請求項1-1に記載の計算マシン。

【請求項1-4】

前記通信バスが、各々が各バイブライン・ユニットと結合されている複数の分歧を備えることと、

第2バスと、

前記分歧の各々、前記バイブルайн・バス・インターフェース、並びに、前記第2バスとそれぞれ結合されたルータと、

を更に含む、請求項1-1に記載の計算マシン。

【請求項1-5】

方法であつて、

通信バスを介して、各々が各ハードウェアに組み込まれたバイブルайнを含む複数のバイブルайн・ユニットの内の第1のものにデータを送信することと、

前記データを前記第1バイブルайн・ユニットで処理することと、

を含む方法。

10

【請求項1-6】

前記データを送信することが、

前記データをルータに送信することと、

前記通信バスの各第1分歧を介して、前記ルータによって前記第1バイブルайн・ユニットに前記データを提供することと、

を含む、請求項1-5に記載の方法。

20

【請求項1-7】

前記データを送信することが、プロセッサによって前記第1バイブルайн・ユニットに前記データを送信することを含む、請求項1-5に記載の方法。

【請求項1-8】

前記データを送信することが、前記複数のバイブルайн・ユニットの第2のものによって前記第1バイブルайнに前記データを送信することを含む、請求項1-5に記載の方法。

【請求項1-9】

前記処理データを前記第1バイブルайн・ユニットによって前記通信バスに駆動することを更に含む、請求項1-5に記載の方法。

【請求項2-0】

前記データを前記第1バイブルайн・ユニットによって処理することが、

ハードウェアに組み込まれたバイブルайн回路によって前記通信バスから前記データを受信することと、

前記ハードウェアに組み込まれたバイブルайн回路によって前記データをメモリにロードすることと、

前記ハードウェアに組み込まれたバイブルайн回路によって前記メモリから前記データを検索することと、

前記ハードウェアに組み込まれたバイブルайн回路によって前記検索データを処理することと、

を含む、請求項1-5に記載の方法。

30

【請求項2-1】

前記データを前記第1バイブルайн・ユニットによって処理することが、

ハードウェアに組み込まれたバイブルайн回路によって前記通信バスから前記データを受信することと、

前記ハードウェアに組み込まれたバイブルайн回路によって前記受信データを処理することと、

前記ハードウェアに組み込まれたバイブルайн回路によって前記処理データをメモリにロードすることと、

前記メモリから前記処理データを検索し、前記ハードウェアに組み込まれたバイブルайн回路によって該処理データを前記通信バスに駆動することと、

40

50

含む、請求項 1～5 に記載の方法。

【請求項 2～2】

前記データを含み且つ前記第 1 バイブライン・ユニットをメッセージの受取人として識別するメッセージを生成すること。

前記データを前記第 1 バイブライン・ユニットに送信することが、前記メッセージから前記第 1 バイブラインが前記メッセージの受取人であることを決定することと
、
を更に含む、請求項 1～5 に記載の方法。

【発明の詳細な説明】

【技術分野】

【0001】

＜優先権の請求＞

この出願は、下記の特許文献 1 に対する優先権を請求するものであり、引用することここで合体させる。

【特許文献 1】米国出願第 60/422,503 号（2002 年 10 月 31 日出願）

【0002】

＜関連出願の相互参照＞

この出願は、「改善された計算アーキテクチャ、関連システム、並びに、方法」と題された下記の特許文献 2、「改善された計算アーキテクチャを有する計算マシン、関連システム、並びに、方法」と題された下記の特許文献 3、「改善された計算アーキテクチャ用バイブルайн加速器、関連システム、並びに、方法」と題された下記の特許文献 4、「プログラマブル回路、関連計算マシン、並びに、方法」と題された下記の特許文献 5 と関連し、これら特許文献は全て 2003 年 10 月 9 日に出願され、共通の所有者を有し、引用することここで合体させる。

【特許文献 2】米国出願第 10/684,102 号

【特許文献 3】米国出願第 10/684,053 号

【特許文献 4】米国出願第 10/683,929 号

【特許文献 5】米国出願第 10/684,057 号

【背景技術】

【0003】

比較的大量のデータを比較的短い期間で処理する通常の計算アーキテクチャは、処理負担を分担する多数の相互接続プロセッサを含む。処理負担を分担することによって、これら多数のプロセッサは、しばしば、所与のクロック周波数で単一プロセッサができるものよりも迅速にデータを処理できる。例えば、これらプロセッサの各々はデータの各部分を処理できるか、或は、処理アルゴリズムの各部分を実行できる。

【0004】

図 1 は、多数プロセッサ・アーキテクチャを有する従来の計算マシン 10 の概略プロック図である。この計算マシン 10 は、マスター・プロセッサ 12 と、相互に通信すると共に該マスター・プロセッサとバス 16 を介して通信する共同プロセッサ 14₁～14_n と、遠隔装置（図 1 では不図示）から生データを受け取る入力ポート 18 と、該遠隔装置に処理データを提供する出力ポート 20 とを含む。また、計算マシン 10 はマスター・プロセッサ 12 に対するメモリ 22 と、共同プロセッサ 14₁～14_n に対する各メモリ 24₁～24_n と、マスター・プロセッサ及び共同プロセッサがバス 16 を介して共有するメモリ 26 を含む。メモリ 22 はマスター・プロセッサ 12 に対するプログラム及び作業メモリの双方の役割を果たし、各メモリ 24₁～24_n は各共同メモリ 14₁～14_n に対するプログラム及び作業メモリの双方の役割を果たす。共有されたメモリ 26 は、マスター・プロセッサ 12 及び共同プロセッサ 14 がそれらの間でデータを転送すること、ポート 18 を介して遠隔装置からデータを転送すること、ポート 20 を介して遠隔装置にデータを転送することを可能としている。またマスター・プロセッサ 12 及び共同プロセッサ 14 は、マシン 10 が生データを処理する速度を制御する共通クロック信号を受け取る。

10

20

30

40

50

【0005】

一般に、計算マシン10は、マスター・プロセッサ12及び共同プロセッサ14の間で生データの処理を効率的に分割する。ソナー・アレイ等の遠隔ソース(図1では不図示)は、ポート18を介して、生データに対する先入れ先出し(FIFO)バッファ(不図示)として作用する共有メモリ26の1つの区分に生データをロードする。マスター・プロセッサ12はバス16を介してメモリ26から生データを検索して、マスター・プロセッサ及び共同プロセッサ14はその生データを処理して、バス16を介して必要に応じてデータをそれらの間に転送する。マスター・プロセッサ12はその処理データを共有メモリ26内に規定された別のFIFOバッファ(不図示)にロードし、遠隔ソースがポート20を介してこのFIFOからその処理データを検索する。

10

【0006】

演算例において、計算マシン10は生データに対するn+1個の各演算を順次実行することによって該生データを処理し、これら演算は一体的に高速フーリエ変換(FFT)等の処理アルゴリズムを構成する。より詳細には、マシン10はマスター・プロセッサ12及び共同プロセッサ14からのデータ・処理バイブラインを形成する。クロック信号の所与の周波数で、こうしたバイブルайнはしばしばマシン10が单一プロセッサのみを有するマシンよりも高速に生データを処理することを可能としている。

【0007】

メモリ26内における生データFIFO(不図示)からの生データ検索後、マスター・プロセッサ12はその生データに対して三角関数等の第1番演算を実行する。この演算は第1番結果を生み出し、それをプロセッサ12がメモリ26内に規定された第1番結果FIFO(不図示)に記憶する。典型的には、プロセッサ12はメモリ22内に記憶されたプログラムを実行し、そのプログラムの制御の下で上述した動作を実行する。プロセッサ12はメモリ22を作業メモリとしても使用し得て、当該プロセッサが第1番演算の中間期間に生成するデータを一時的に記憶する。

20

【0008】

次に、メモリ26内における第1番結果FIFO(不図示)からの第1番結果検索後、共同プロセッサ14₁はその第1番結果に対して対数関数等の第2番演算を実行する。この第2番演算は第2番結果を生み出し、それを共同プロセッサ14₁がメモリ26内に規定された第2番結果FIFO(不図示)に記憶する。典型的には、共同プロセッサ14₁はメモリ24₁内に記憶されたプログラムを実行し、そのプログラムの制御の下で上述した動作を実行する。共同プロセッサ14₁はメモリ24₁を作業メモリとしても使用し得て、当該共同プロセッサが第2番演算の中間期間に生成するデータを一時的に記憶する。

30

【0009】

次に共同プロセッサ24₂～24_nは、共同プロセッサ24₁に対して先に議論されたものと同様に、(第2番結果～第(n-1)番)結果に対して(第3番演算～第n番)演算を順次実行する。

【0010】

共同プロセッサ24_nによって実行される第n番演算は最終結果、即ち処理データを生み出す。共同プロセッサ24_nはその処理データをメモリ26内に規定された処理データ FIFO(不図示)内にロードし、遠隔装置(図1では不図示)がこのFIFOからその処理データを検索する。

40

【0011】

マスター・プロセッサ12及び共同プロセッサ14は処理アルゴリズムの種々の演算を同時に実行するので、計算マシン10は、しばしば、種々の演算を順次実行する单一プロセッサを有する計算マシンよりも生データを高速に処理することができる。詳細には、单一プロセッサは、生データから成る先行集合に対する全(n+1)個の演算を実行するまで、生データから成る新しい集合を検索できない。しかし、以上に議論したバイブルайн技術を用いて、マスター・プロセッサ12は第1演算だけを実行後に生データから成る新しい集合を検索できる。結果として、所与のクロック周波数でこのバイブルайн技術は、

50

单一プロセッサ・マシン（図1では不図示）と比較して約 $n+1$ 倍だけマシン1.0が生データを処理する速度を増大することができる。

【0012】

代替的には、計算マシン1.0は、生データに対するFFT等の処理アルゴリズムの（ $n+1$ ）例を同時に実行することによって該生データを並列して処理し得る。即ち、もしそのアルゴリズムが先行する例において先に記載されたような（ $n+1$ ）個の順次演算を含めば、マスター・プロセッサ1.2及び共同プロセッサ1.4の各々は生データからそれぞれが成る各集合に対して、順次、全（ $n+1$ ）個の演算を実行する。その結果として、所与のクロック周波数で、先のバイブライン技術と同様のこの並列処理技術は、单一プロセッサ・マシン（図1では不図示）と比較して約 $n+1$ 倍だけマシン1.0が生データを処理する速度を増大することができる。
10

【0013】

残念ながら、計算マシン1.0は单一プロセッサ・計算マシン（図1では不図示）と比べてより迅速にデータを処理できるが、マシン1.0のデータ処理速度はしばしばプロセッサ・クロックの周波数より非常に小さい。詳細には、計算マシン1.0のデータ処理速度はマスター・プロセッサ1.2及び共同プロセッサ1.4がデータ処理するのに必要な時間によって制限される。簡略化のため、この速度制限の例はマスター・プロセッサ1.2と連携して議論されているが、この議論は共同プロセッサ1.4にも適用されることを理解して頂きたい。先に議論されたように、マスター・プロセッサ1.2は所望の方式でデータを操作すべくプロセッサを制御するプログラムを実行する。このプログラムはプロセッサ1.2が実行する複数の命令から成るシーケンスを含む。残念ながら、プロセッサ1.2は典型的には單一命令を実行するために多数のクロック・サイクルを必要とし、そしてしばしばデータの單一値を処理すべく多数の命令を実行しなければならない。例えば、プロセッサ1.2が第1データ値A（不図示）を第2データ値B（不図示）で乗算することを假定する。第1クロック・サイクル中、プロセッサ1.2はメモリ2.2から乗算命令を検索する。第2及び第3クロック・サイクル中、プロセッサ1.2はメモリ2.6からA及びBをそれぞれ検索する。第4クロック・サイクル中、プロセッサ1.2はA及びBを乗算し、そして第5クロック・サイクル中に結果としての積をメモリ2.2或は2.6に記憶するか、或は、その結果としての積を遠隔装置（不図示）に提供する。これは最良ケースのシナリオであり、その理由は多くの場合にプロセッサ1.2はカウンタの初期化及び明細等のオーバーヘッド・タスクに対して付加的なクロック・サイクルを必要とするからである。それ故に、よくてもプロセッサ1.2はA及びBを処理すべく5クロック・サイクルを必要とするか、或は、1データ値当たり平均2.5クロック・サイクルを必要とする。
20
30

【0014】

結果として、計算マシン1.0がデータを処理する速度は、しばしば、マスター・プロセッサ1.2及び共同プロセッサ1.4を駆動するクロックの周波数より非常に低い。例えば、もしプロセッサ1.2は1.0ギガヘルツ（GHz）でクロックされるが、1データ値当たり平均2.5クロック・サイクルを必要とすれば、効果的なデータ処理速度は（1.0GHz）/2.5=0.4GHzと同等である。この効果的なデータ処理速度は、しばしば、1秒当たり演算数の単位で特徴付けされる。それ故に、この例において、1.0GHzのクロック速度で、プロセッサ1.2は0.4ギガ演算数/秒（Gops）で使用限界が定められる。
40

【0015】

図2は、所与クロック周波数で且つしばしば該バイブルайнがクロックされる速度と略同一速度で、プロセッサが可能であるよりは高速で典型的にはデータを処理できるハードウェアに組み込まれたデータ・バイブルайн3.0のプロック線図である。バイブルайн3.0は、プログラム命令を実行することなく、各データに対する各演算を各々が実行する演算子回路3.2₁～3.2_nを含む。即ち、所望の演算は回路3.2内に「書き込み」が為されて、それがプログラム命令の必要性なしに自動的にその演算を具現化するよう為す。プログラム命令の実行と関連されたオーバーヘッドを減ずることによって、バイブルайн3.0
50

は所与のクロック周波数でプロセッサが可能であるよりは単位秒当たりより多くの演算を典型的には実行する。

【0016】

例えば、バイブライン30は所与のクロック周波数でプロセッサが可能であるよりは高速で以下の式¹をしばしば解くことができる。

$$Y(x_1) = (5x_1 + 3) 2^{*k}$$

ここで、 x_1 は複数の生データ値から成るシーケンスを表す。この例において、演算子回路32₁は $5x_1$ を計算する乗算器であり、回路32₂は $5x_1 + 3$ を計算する加算器であり、そして回路32_n(n=3)は $(5x_1 + 3) 2^{*k}$ を計算する乗算器である。

【0017】

第1クロック・サイクルk=1中、回路32₁はデータ値 x_1 を受け取って、それを5で乗じて、 $5x_1$ を生成する。

【0018】

第2クロック・サイクルk=2中、回路32₂は回路32₁から $5x_1$ を受け取って、3を加えて、 $5x_1 + 3$ を生成する。またこの第2クロック・サイクル中に回路32₁は $5x_2$ を生成する。

【0019】

第3クロック・サイクルk=3中、回路32₃は回路32₂から $5x_2 + 3$ を受け取って、 2^{*k} で乗じて(効果としては、 x_1 だけ $5x_1 + 3$ を左シフトする)、第1結果 $(5x_1 + 3) 2^{*k}$ を生成する。またこの第3クロック・サイクル中に回路32₁は $5x_3$ を生成し、回路32₂は $5x_3 + 3$ を生成する。

【0020】

このようにしてバイブルайн30は、全ての生データ値が処理されるまで、引き続く生データ値 x_k の処理を続行する。

【0021】

結果として、生データ値 x_1 の受け取り後の2つのクロック・サイクルの遅延、即ち、この遅延はバイブルайн30の待ち時間としばしば呼称され、バイブルайнは結果 $(5x_1 + 3) 2^{*k}$ を生成し、その後、1つの結果を生成する、即ち各クロック・サイクル毎に $(5x_2 + 3) 2^{*k}$ 、 $(5x_3 + 3) 2^{*k}$ 、 \dots 、 $(5x_n + 3) 2^{*k}$ を生成する。

【0022】

待ち時間を無視して、バイブルайн30はこうしてクロック速度と同等のデータ処理速度を有する。比較して、マスター・プロセッサ12及び共同プロセッサ14(図1)が先の例におけるようにクロック速度の0.4倍であるデータ処理速度を有すると仮定すれば、バイブルайн30は、所与のクロック速度で、計算マシン10(図1)よりも2.5倍高速でデータを処理できる。

【0023】

更に図2で参照されるように、設計者はフィールド・プログラマブル・ゲート・アレイ(FPGA)等のプログラマブル・ロジックIC(PЛИC)にバイブルайн30を具現化することを選択可能があり、その理由はPлиCが特殊用途IC(ASiC)が為すよりも多くの設計及び変更の柔軟性を許容するからである。PлиC内にハードウェアに組み込まれた接続を構成するため、設計者はPлиC内に配置された相互接続構成レジスタを單に所定バイナリー状態に設定する。全てのこうしたバイナリー状態の組み合せはしばしば「ファームウェア」と呼称される。典型的には、設計者はこのファームウェアをPлиCと結合された不揮発性メモリ(図2では不図示)内にロードする。PлиCを「ターンオン」すると、それはファームウェアをそのメモリから相互接続構成レジスタにダウンロードする。それ故に、PлиCの機能を変更すべく、設計者は單にそのファームウェアを変更して、PлиCがその変更されたファームウェアを相互接続構成レジスタにダウンロードすることを可能とする。ファームウェアを單に変更することによってPлиCを変更する能力は、モデル作成段階中や「フィールド内」にバイブルайн30をアップグレードするために特に有用である。

10

20

30

40

50

【0024】

10 残念ながら、ハードウェアに組み込まれたバイブルайн30は重要な意思決定、特に入れ子意思決定を引き起こすアルゴリズムを実行すべき最良の選択でない可能性がある。プロセッサは、典型的には、入れ子意思決定命令（例えば、「もしAであれば、Bを為し、またもしCであれば、Dを為し、・・・またnを為し等々」のように、入れ子条件命令）を、比肩する長さの演算命令（例えば、「A+B」）を実行できる程に高速に実行できる。しかしバイブルайн30は、比較的単純な決定（例えば、「A>B？」）を効率的に為し得るが、典型的にはプロセッサができる程に効率的に入れ子決定（例えば、「もしAであれば、Bを為し、またもしCであれば、Dを為し、・・・またnを為す」）を為すことができない。この非効率性の1つの理由は、バイブルайн30はほんの僅かなオンボード・メモリしか持たないことがあり、したがって外部作業／プログラム・メモリ（不図示）にアクセスすることを必要とすることがあるからである。そして、こうした入れ子決定を実行すべくバイブルайн30を設計することができるが、必要とされる回路のサイズ及び複雑性はしばしばそうした設計を非現実的に為し、特にアルゴリズムが多数の種々の入れ子決定を含む場合にそうである。

【0025】

結果として、プロセッサは典型的には重要な意思決定を必要とする用途において使用され、ハードウェアに組み込まれたバイブルайнは殆ど意思決定が為されないか或は意思決定されない「ナンバークランチング（数値データ処理）」用途に典型的には限定される。

【0026】

更には、下記に議論されるように、典型的には、特にバイブルайн30が多数のPLLを含む場合、図2のバイブルайн30等のハードウェアに組み込まれたバイブルайнを設計／変更するよりも、図1の計算マシン10等のプロセッサに基づく計算マシンを設計／変更することが非常に易しい。

【0027】

プロセッサ及びそれらの周辺機器（例えば、メモリ）等の計算構成要素は、典型的には、プロセッサに基づく計算マシンを形成すべくそれら構成要素の相互接続を補助する工業規格通信インターフェースを含む。

【0028】

典型的には、規格通信インターフェースは2つの層、即ち、物理層及びサービス層を含む。

【0029】

物理層は、回路とこの回路のインターフェース及び動作パラメータを形成する対応回路相互接続とを含む。例えば、物理層はそれら構成要素を1つのバスに接続するピンと、それらのピンから受け取ったデータをラッチするバッファと、信号をそれらピンに駆動するドライバとを含む。動作パラメータは、ピンが受け取るデータ信号の許容可能電圧範囲と、データの書き込み及び読み取りのための信号タイミングと、動作の支擱されたモード（例えば、バーストモード、ページモード）とを含む。従来の物理層はトランジスタ-トランジスタ論理（TTL）及びRAMBUSを含む。

【0030】

サービス層は、計算構成要素のデータ転送のためのプロトコルを含む。このプロトコルはデータのフォーマットと、構成要素によるフォーマット済みデータの送受信の方法とを含む。従来の通信プロトコルは、ファイル転送プロトコル（FTP）及び伝送制御プロトコル／インターネット・プロトコル（TCP/IP）を含む。

【0031】

結果として、製造業者やその他は工業規格通信インターフェースを有する計算構成要素を典型的には設定するので、そうした構成要素のインターフェースを典型的には設計できて、それを他の計算構成要素と比較的少ない労力で相互接続することができる。これは、計算マシンの他の部分の設計に設計者自信の時間を省くことを可能として、各種構成要素を追加或は除去することによってそのマシンを変更することを可能としている。

【0032】

工業規格通信インターフェースを支援する計算構成要素を設計することは、設計ライブラリから既存の物理層を用いることによって設計時間を節約することを可能としている。これは、設計者が構成要素を既製の計算構成要素と容易にインターフェースすることを保證するものである。

【0033】

そして、共通した工業規格通信インターフェースを支援する計算構成要素を用いる計算マシンを設計することは、設計者がそれら構成要素を少しの時間及び労力で相互接続することを可能としている。それら構成要素は共通インターフェースを支援するので、設計者はそれらをシステム・バスを介して少しの設計労力で相互接続することができる。そして、その支援されたインターフェースは工業規格であるので、マシンを容易に変更することができる。例えば、システム設計が進化するに伴って種々の構成要素及び周辺機器をマシンに追加することができるか、或は、テクノロジーが進化するに伴って次世代の構成要素を追加／設計することが可能である。更には、構成要素が通常の工業規格サービス層を支援するので、計算マシンのソフトウェアに対応するプロトコルを具現化する既存のソフトウェア・モジュールを組み込むことができる。それ故に、インターフェース設計が本質的には既に整っているので少しの労力で構成要素をインターフェースでき、よって、マシンに所望の機能を実行させるマシンの各種部分（例えばソフトウェア）の設計に集中することができる。

10

【0034】

しかし残念ながら、図2のパイプライン30等のハードウェアに組み込まれたパイプラインを形成すべく、使用されるPLC等の各種構成要素に対する既知の工業規格サービス層が全くない。

20

【0035】

結果として、多数のPLCを有するパイプラインを設計すべく、多大な時間を費やし、「ゼロから」設計し且つ種々のPLCの間の通信インターフェースのサービス層をデバッグする多大な労力を行使する。典型的には、そうしたその場限りのサービス層は種々のPLC間で転送されるデータのパラメータに依存する。同じように、プロセッサとインターフェースするパイプラインを設計すべく、パイプライン及びプロセッサの間の通信インターフェースのサービス層の設計及びデバッグに関して多大な時間を費やし且つ多大な労力を行使する必要がある。

30

【0036】

同様に、こうしたパイプラインをPLCを該パイプラインに追加することによって変更すべく、典型的には、その追加されたPLCと既存のPLCとの間の通信インターフェースのサービス層の設計及びデバッグに関して多大な時間を費やし且つ多大な労力を行使する。同じように、プロセッサを追加することによってパイプラインを変更すべく、或は、パイプラインを追加することによって計算マシンを変更すべく、パイプライン及びプロセッサの間の通信インターフェースのサービス層の設計及びデバッグに関して多大な時間を費やし且つ多大な労力を行使しなければならないであろう。

40

【0037】

結果として、図1及び図2で参照されるように、多数のPLCをインターフェースすることとプロセッサをパイプラインにインターフェースすることとの難しさのため、計算マシンを設計する際に多大な妥協を為すことがしばしば強いられる。例えば、プロセッサに基づく計算マシンでは、ナンバークランチング速度を、複雑な意思決定を為す能力に対する設計／変更の柔軟性と交換することを強いられる。逆に、ハードウェアに組み込まれたパイプラインに基づく計算マシンでは、複雑な意思決定を為す能力と設計／変更の柔軟性を、ナンバークランチング速度と交換することを強いられる。更には、多数のPLCをインターフェースすることに関する難しさのため、少數のPLCよりも多くのPLCを有するパイプラインに基づくマシンを設計することはしばしば実際的ではない。その結果、実際的なパイプラインに基づくマシンはしばしば制限された機能しか有しない。そ

50

して、プロセッサを P L I C とインターフェースすることに関する難しさのため、プロセッサを 1 つの P L I C より多くの P L I C にインターフェースすることは実際的ではない。その結果、プロセッサ及びパイプラインを組み合わせることによって獲得される利益は最少となる。

【発明の開示】

【発明が解決しようとする課題】

【0038】

それ故に、プロセッサに基づくマシンの意思決定を為す能効を、ハードウェアに組み込まれたパイプラインに基づくマシンのナンバークランチング速度と組み合わせることを可能とする新しい計算アーキテクチャに対する要望が生じてきている。

10

【課題を解決するための手段】

【0039】

本発明の実施例に従えば、パイプライン加速器はバスと、各々がそのバスと結合されると共に少なくとも 1 つのハードウェアに組み込まれたパイプライン回路を含んでいる複数のパイプライン・ユニットとを含む。

【0040】

複数のパイプライン・ユニットをパイプライン加速器に含ませることによって、單一パイプライン加速器と比較して加速器のデータ処理性能を増大することができる。更には、相互に通信すると共に他のビアと共にバスを介して通信するようにパイプライン・ユニットを設計することによって、パイプライン・ユニットの数を変えて、パイプラインのバスに対する單なる結合或は結合解除によって加速器のコンフィギュレーション及び機能を変えることができる。これは、パイプライン・ユニットの 1 つを変えるたび、或は、加速器内のパイプライン・ユニットの数を変えるたびに、パイプライン・ユニット・インターフェースを設計せねば再設計することなくするものである。

20

【発明を実施するための最良の形態】

【0041】

図 3 は、本発明の一実施例に従ったビア-ベクトル・アーキテクチャを有する計算マシン 40 の概略プロック線図である。ホストプロセッサ 42 に加えて、ビア-ベクトル・マシン 40 はパイプライン加速器 44 を含み、それがデータ処理の少なくとも一部を実行して、図 1 の計算マシン 10 における共同プロセッサ 14 の列と効果的に置き換わる。それ故に、ホストプロセッサ 42 及び加速器 44（又は以下に繰り返されるようにそのユニット）はデータ・ベクトルを前後に転送できる「ビア」である。加速器 44 はプログラム命令を実行しないので、所与のクロック周波数で共同プロセッサの列ができるよりも著しく高速にデータに対して数学的に集中的な演算を典型的には実行する。結果として、プロセッサ 42 の意思決定能力と加速器 44 のナンバークランチング能力とを組み合わせることによって、マシン 40 はマシン 10 等の従来の計算マシンと同一の能力を有するが、しばしばそれよりもデータをより高速に処理することができる。更には、以下に議論されるように、加速器 44 にホストプロセッサ 42 の通信インターフェースと互換性がある通信インターフェースを設けることが、特にプロセッサの通信インターフェースが工業規格である場合に、マシン 40 の設計及び変更を補助する。そして、加速器 44 が多数のパイプライン・ユニット（例えば、P L I C に基づく回路）を含む場合、それら各ユニットに同一の通信インターフェースを設けることが、特にそれら通信インターフェースが工業規格インターフェースと互換性がある場合に、当該加速器の設計及び変更を補助する。更には、マシン 40 は以下に議論されると共に先行して引用された特許出願におけるような他の長所等をも提供し得る。

30

【0042】

更に図 3 で参照されるように、ホストプロセッサ 42 及びパイプライン加速器 44 に加えて、ビア-ベクトル・計算マシン 40 は、プロセッサ・メモリ 46、インターフェース・メモリ 48、パイプライン・バス 50、1 つ或はそれ以上のファームウェア・メモリ 52、任意選択的な生データ入力ポート 54、処理済みデータ出力ポート 58、並びに、任

40

50

意選択的なルータ 6-1 を含む。

【0043】

ホストプロセッサ 4-2 は処理ユニット 6-2 及びメッセージ・ハンドラー 6-4 を含み、プロセッサ・メモリ 4-6 は処理ユニット・メモリ 6-6 及びハンドラー・メモリ 6-8 を含み、そのそれぞれがプロセッサ・ユニット及びメッセージ・ハンドラーに対するプログラム及び作業の両メモリとして役立っている。プロセッサ・メモリ 4-6 は、加速器コンフィギュレーション・レジストリ 7-0 及びメッセージ・コンフィギュレーション・レジストリ 7-2 をも含み、それらが、ホストプロセッサ 4-2 が加速器 4-4 の機能を構成すると共に、該ホストプロセッサ 4-2 がメッセージ・ハンドラー 6-4 が送信及び受信するメッセージのフォーマットを構成することを可能とするそれぞれのコンフィギュレーション・データを記憶する。

10

【0044】

パイプライン加速器 4-4 は少なくとも 1 つの P-L-I-C (図 4) 上に配置され、プログラム命令を実行することなしに各データを処理するハードウェアに組み込まれたパイプライン 7-4-1-7-4-2 を含む。ファームウェア・メモリ 5-2 は加速器 4-4 に対するコンフィギュレーション・ファームウェアを記憶する。もし加速器 4-4 が多数の P-L-I-C 上に配置されたら、それら P-L-I-C 及びそれらの各ファームウェア・メモリは多数パイプライン・ユニット内に配置され得るが、それらパイプラインは図 4 乃至図 8 と連携して以降に更に議論される。代替的には、加速器 4-4 は少なくとも 1 つの A-S-I-C 上に配置され得て、その A-S-I-C がひとたび形成されたならば構成不可能である内部相互接続を有し得る。この代替例において、マシン 4-0 はファームウェア・メモリ 5-2 を省略し得る。更には、加速器 4-4 が多数パイプライン 7-4-1-7-4-2 を含んで示されているが、ただ 1 つのパイプラインを含み得る。加えて、図示されていないが、加速器 4-4 はディジタル信号プロセッサ (D-S-P) 等の 1 つ或はそれ以上のプロセッサを含み得る。更には、図示されていないが、加速器 4-4 はデータ入力ポート及び／或はデータ出力ポートを含み得る。

20

【0045】

そしてホストプロセッサ 4-2 及びパイプライン加速器 4-4 は種々の I-C 上に配置されているように議論されるが、ホストプロセッサ及びパイプライン加速器は同一の I-C 上に配置され得る。

30

【0046】

ピア-ベクトル・マシン 4-0 の一般動作は、先行して引用された「改善された計算アーキテクチャ、関連システム、並びに、方法」と題された特許文献 2 に議論されており、ホストプロセッサ 4-2 の構造及び動作は、先行して引用された「改善された計算アーキテクチャを有する計算マシン、関連システム、並びに、方法」と題された特許文献 3 に議論されており、パイプライン加速器 4-4 の構造及び動作は、先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献 4 と図 4 乃至図 8 と連携された以下に議論されている。

【0047】

図 4 は、本発明の一実施例に従った図 3 のパイプライン加速器 4-4 のユニット 7-8 のブロック線図である。

40

【0048】

加速器 4-4 は 1 つ或はそれ以上のそうしたパイプライン・ユニット 7-8 を含み (図 4 に 1 つのみ示される)、それらの各々は P-L-I-C 或は A-S-I-C 等のパイプライン回路 8-0 を含む。以下で更に議論されると共に先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献 4 にあるように、各パイプライン・ユニット 7-8 はホストプロセッサ 4-2 (図 3) の「ピア」であると共に加速器 4-4 の他のパイプライン・ユニットの「ピア」である。即ち、各パイプライン・ユニット 7-8 はホストプロセッサ 4-2 或は他の任意のパイプライン・ユニットと直接通信できる。よって、このピア-ベクトル・アーキテクチャは、もしパイプライン・ユニット 7-8 の全てがマスター-パイプライン・ユニット (不図示) 或はホストプロセッサ 4-2

50

等の中央箇所を通じて通信した場合に生ずることとなるデータ「ボトルネック」を防止する。更にはこのアーキテクチャは、マシンに対する重大な変更なしに、ピア-ペクトル・マシン 4.0 (図 3) からピアを追加するか或はピアを除去することを可能とする。

【0049】

バイブルイン回路 8.0 は通信インターフェース 8.2 を含み、それが、ホストプロセッサ 4.2 (図 3) 等のピアと、通信シェル 8.4 を介したハードウェアに組み込まれたバイブルイン 7.4₁～7.4_n、バイブルイン・コントローラ 8.6、例外マネージャ 8.8、並びに、コンフィギュレーション・マネージャ 9.0 等の、バイブルインの他の構成要素との間でデータを転送する。バイブルイン回路 8.0 は工業規格バス・インターフェース 9.1 及び通信バス 9.3 をも含み得て、インターフェース 8.2 をインターフェース 9.1 と接続する。代替的には、インターフェース 9.1 の機能は通信インターフェース 8.2 内に含まれ得て、バス 9.3 は省略されえる。

【0050】

バイブルイン回路 8.0 の複数の構成要素を複数の個別モジュールとして設計することによって、そのバイブルイン回路の設計をしばしば簡略化することができる。即ち、それら構成要素の各々を個別に設計及び試験することができ、次いでそれらを統合するものであり、それはソフトウェア或はプロセッサに基づく計算システム (図 1 のシステム 1.0 等) を設計する際に用いられることと非常に似ている。加えて、これら構成要素、特に設計者が他のバイブルイン設計においてたぶん頻繁に使用するような通信インターフェース 8.2 等の構成要素を規定するハードウェア記述言語 (HDL) をライブラリ (不図示) 内に記憶でき、よって同一構成要素を使用する将来のバイブルイン設計の設計及び試験の時間を低減する。即ち、ライブラリから HDL を使用することによって、設計者はスクラッチから先行して具現化された構成要素を再設計する必要性がなく、よって設計者の努力を先行して具現化されていない構成要素の設計に対して、或は、先行して具現化された構成要素の変更に対して集中できる。更には、ライブラリ内にバイブルイン回路 8.0 の多数バージョン或はバイブルイン加速器 4.4 の多数バージョンを生成する HDL を記憶できて、既存の設計の中から精選及び選択できるようになる。

【0051】

図 4 で更に参照されるように、通信インターフェース 8.2 はメッセージ・ハンドラー 6.4 (図 3) によって認識されるフォーマットで (ここではバス・インターフェース 9.1 を介して) データを送受信し、よってピア-ペクトル・マシン 4.0 (図 3) の設計及び変更を典型的には補助する。例えば、もしデータ・フォーマットが高速 1/0 フォーマット等の工業規格であれば、ホストプロセッサ 4.2 及びバイブルイン・ユニット 7.8 の間にカスタムインターフェースを設計する必要がない。更には、バイブルイン・ユニット 7.8 を非バス・インターフェースの代わりにバイブルイン・バス 5.0 を介してホストプロセッサ 4.2 (図 3) 等の他のピアと通信させることを可能することによって、バイブルイン・ユニットが追加或は除去されたたびにスクラッチから非バス・インターフェースを再設計する代わりにバイブルイン・バスにそれら (又はそれらを保持する回路カード) を単に接続或は接続解除することによってバイブルイン・ユニットの数を変更できる。

【0052】

ハードウェアに組み込まれたバイブルイン 7.4₁～7.4_n は図 3 と並んで先に議論されたように且つ先行して引用された「改善された計算アーキテクチャ、関連システム、並びに、方法」と題された特許文献 2 や「改善された計算アーキテクチャ用バイブルイン加速器、関連システム、並びに、方法」と題された特許文献 4 におけるように、データに対する各演算を実行し、通信シェル 8.4 はバイブルインをバイブルイン回路 8.0 の他の構成要素やバイブルイン・ユニット 7.8 の他の回路 (以下に議論されるデータ・メモリ 9.2 等) にインターフェースする。

【0053】

コントローラ 8.6 は SYNC 値号や他のピアからの特別なバイブルイン-バス通信 (即ち、「事象」) に応じてハードウェアに組み込まれたバイブルイン 7.4₁～7.4_n を同期さ

10

20

30

40

50

等、パイプラインがそれらの各データ演算を実行するシーケンスをモニタし制御する。例えば、ホストプロセッサ42等のピアはSYNC信号を脈動するか、パイプライン・ユニット78に事象をパイプライン・バス50を介して送信して、ピアがデータ・ブロックをパイプライン・ユニットに送信し終えたことを示し、そして、ハードウェアに組み込まれたパイプライン74₁～74₆にこのデータを処理し始めさせる。典型的にはSYNC信号が使用されてタイムクリティカルな演算を同期し、事象が使用されて、非タイムクリティカルな演算を同期する。典型的には、事象はしばしば「ドアベル」と呼称されるデータ無し通信である。しかし事象はデータを含み得て、その場合しばしば「事象メッセージ」と呼称される。SYNC信号及び事象は、先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献4に更に議論されている。10

【0054】

例外マネージャ88はハードウェアに組み込まれたパイプライン74₁～74₆、通信インターフェース82、通信シェル84、コントローラ86、並びに、バス・インターフェース91（もしあれば）の状況をモニタし、ホストプロセッサ42（図3）に例外を報告する。例えば、もし通信インターフェース82におけるバッファがオーバーフローすれば、例外マネージャ88はこれをホストプロセッサ42に報告する。例外マネージャはその例外を生んだ問題を修正するか成はその修正を試みることも可能である。例えば、オーバーフローしているバッファに対して例外マネージャ88は、直接的或は以下に議論されるようなコンフィギュレーション・マネージャ90を介して、そのバッファのサイズを増大し得る。20

【0055】

コンフィギュレーション・マネージャ90はハードウェアに組み込まれたパイプライン74₁～74₆、通信インターフェース82、通信シェル84、コントローラ86、例外マネージャ88、並びに、インターフェース91（もしあれば）のソフト・コンフィギュレーションを、ホストプロセッサ42（図3）からのソフト・コンフィギュレーション・データに応じて設定し、これは先に引用された「改善された計算アーキテクチャ、関連システム、並びに、方法」と題された特許文献2に議論され、ハード・コンフィギュレーションはパイプライン回路80のトランジスタ及び回路ブロックのレベル上における実際のトポロジーを示し、ソフト・コンフィギュレーションはハード構成された構成要素の物理的バラメータ（例えば、データ幅、テーブル・サイズ）を示す。即ち、ソフト・コンフィギュレーション・データはプロセッサ（図4に不図示）のレジスタにロードされ得るプロセッサの動作モード（例えば、バースト・メモリ・モード）を設定するデータと同様である。例えばホストプロセッサ42は、コンフィギュレーション・マネージャ90に通信インターフェース82におけるデータ及び事象キューの数及び各優先レベルを設定させるソフト・コンフィギュレーション・データを送信し得る。例外マネージャ88は、コンフィギュレーション・マネージャ90に、例えば、通信インターフェース82におけるオーバーフローしているバッファのサイズを増大させるソフト・コンフィギュレーション・データをも送信し得る。30

【0056】

工業規格バス・インターフェース91は、通信インターフェース82から幾つかのインターフェース回路を効率的にオフロードすることによって、通信インターフェース82のサイズ及び複雑性を低減する従来のバス・インターフェース回路である。それ故に、もしパイプライン・バス50或はルータ61（図3）のバラメータを変更することを望めば、インターフェース91を変更するだけでなく、通信インターフェース82を変更する必要がない。代替的には、パイプライン回路80の外部であるIC（不図示）内にインターフェース91を配置し得る。パイプライン回路80からインターフェース91をオフロードすることは、例えばハードウェアに組み込まれたパイプライン74₁～74₆及びコントローラ86の用途のパイプライン回路上のリソースを解放する。或は、先に議論されたように、バス・インターフェース91は通信インターフェース82の一部であり得る。40

JP 2006-515941 A 2006.6.8

20

30

40

50

【0057】

図4で更に参照されるように、バイブライン回路80に加えて、加速器44のバイブルайн・ユニット78はデータ・メモリ92を含み、そしてもしバイブルайн回路がPLI-Cであれば、ファームウェア・メモリ52を含む。

【0058】

データ・メモリ92は、データがホストプロセッサ42(図3)等の別のピアとハードウェアに組み込まれたバイブルайн74₁～74_nとの間を流れる時にそのデータをバッファするものであり、ハードウェアに組み込まれたバイブルайнに対する作業メモリでもある。通信インターフェース82はデータ・メモリ92を(通信バス94及びもしあれば工業規格インターフェース91を介して)バイブルайн・バス50とインターフェースし、通信シェル84はそのデータ・メモリをハードウェア・バイブルайн74₁～74_nとインターフェースする。
10

【0059】

データ・メモリ92(或はバイブルайн・ユニット78の他の部分)はバイブルайн・ユニットのプロファイルをも記憶し得る。そのプロファイルはホストプロセッサ42(図3)に対してバイブルайн・ユニット78を充分に記述して、それ自体、バイブルайн・ユニット、ピア・ベクトル・マシン40(図3)の他のピアを相互通信用に適切に構成する。例えば、プロファイルはバイブルайн・ユニット78が履行することができるデータ演算や通信プロトコルを識別できる。結果として、ピア・ベクトル・マシン40の初期化中にプロファイルを読むことによって、ホストプロセッサ42はメッセージ・ハンドラー64(図3)を適切に構成できて、バイブルайн・ユニット78との通信を為す。この技術は「プラグ・アンド・プレイ」技術と類似しており、それによってコンピュータはそれ自体を構成できて、ディスク・ドライブ等の新しくインストールされた周辺機器と通信する。ホストプロセッサ42及びバイブルайн・ユニット78のコンフィギュレーションは、先行して引用された「改善された計算アーキテクチャ、関連システム、並びに、方法」と題された特許文献2や、「プログラマブル回路、関連計算マシン、並びに、方法」と題された特許文献5に更に議論されている。
20

【0060】

バイブルайн回路80がPLI-Cである図3と連携されて先に議論されたように、ファームウェア・メモリ52はバイブルайн回路のハード・コンフィギュレーションを設定するファームウェアを記憶する。このメモリ52はファームウェアを加速器44の構成中にバイブルайн回路80にロードし、加速器44の構成中或はその後に通信インターフェース82を介してホストプロセッサ42(図3)から変更されたファームウェアを受信し得る。ファームウェアのローディング及び受信は、先行して引用された「改善された計算アーキテクチャ、関連システム、並びに、方法」と題された特許文献2、「改善された計算アーキテクチャを有する計算マシン、関連システム、並びに、方法」と題された特許文献3、「プログラマブル回路、関連計算マシン、並びに、方法」と題された特許文献5に更に議論されている。
30

【0061】

図4で更に参照されるように、バイブルайн・ユニット78は回路ボード或はカード98を含み得て、その上にバイブルайн回路80、データ・メモリ92、並びに、ファームウェア・メモリ52が配置されている。この回路ボード98は、ドーター・カードがパーソナルコンピュータ(不図示)のマザーボードのスロット内にプラグインされ得ることと非常によく似て、バイブルайн・バス・コネクタ(不図示)内にプラグインされ得る。図示されていないが、バイブルайн・ユニット78は従来のI-Cや電力調整器及び電力シーケンサ等の構成要素を含み得て、これらI-C／構成要素も公知であるようにカード98上に配置され得る。
40

【0062】

バイブルайн・ユニット78の構造及び動作の更なる詳細は、先行して引用された「改善された計算アーキテクチャ用バイブルайн加速器、関連システム、並びに、方法」と題
50

された特許文献4に議論されている。

【0063】

図5は、本発明の別の実施例に従った図3のパイプライン加速器44のパイプライン・ユニット100のブロック線図である。パイプライン・ユニット100は、そのパイプライン100が多数のパイプライン回路80。ここでは2つのパイプライン回路80a及び80bを含むことを除いて、図4のパイプライン・ユニット78と類似している。パイプライン回路80の数を増大することは、典型的には、ハードウェアに組み込まれたパイプライン74₁～74_nの数nの増大、よってパイプライン・ユニット78と比較してのパイプライン・ユニット100の機能に関する増大を可能とする。更には、パイプライン・ユニット100はパイプライン回路80aに対するファームウェア・メモリ52aとパイプライン回路80bに対するファームウェア・メモリ52bとを含む。代替的には、パイプライン回路80a及び80bは单一ファームウェア・メモリを共有し得る。
10

【0064】

パイプライン・ユニット100において、サービス構成要素、即ち、通信インターフェース82、コントローラ86、側面マネージャ88、コンフィギュレーション・マネージャ90、並びに、任意選択的な工業規格バス・インターフェース91はパイプライン回路80a上に配置され、パイプライン74₁～74_n及び通信シェル84はパイプライン回路80b上に配置される。サービス構成要素及びパイプライン74₁～74_nを個別のパイプライン回路80a及び80b上に位置決めすることによって、サービス構成要素及びパイプラインが同一パイプライン回路上に位置決めされる場合で可能であるものよりも、より多くの数nのパイプライン及び/又はより複雑なパイプラインを含むことができる。代替的には、パイプライン74₁～74_nをインターフェース82及びコントローラ86にインターフェースする通信シェル84の部分はパイプライン回路80a上に配置され得る。
20

【0065】

パイプライン・ユニット100の構造及び動作の異なる詳細は、先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献4に議論されている。

【0066】

図6は、本発明の実施例に従った多数のパイプライン・ユニット78（図4）或は100（図5）を有する図3の加速器44のブロック線図である。説明の簡略化のために、加速器44は多数のパイプライン・ユニット78₁～78_nを有するように議論されるが、該加速器が多数のパイプライン・ユニット100或はユニット78及び100の組み合せを含み得ることを理解して頂きたい。多数のパイプライン・ユニット78を含むことによって、ただ1つのパイプライン・ユニットを有する加速器と比較して加速器44の機能及び処理能力を増大できる。更には、各パイプライン・ユニット78が典型的には一般的な工業規格インターフェースを有するので、パイプライン・ユニットを追加或は削除することで加速器44を容易に変更できる。
30

【0067】

多数のパイプライン加速器44の1つの具現化例において、工業規格バス・インターフェース91は各パイプライン・ユニット78₁～78_nから省略され、單一の（パイプライン・ユニットに対して）外部のインターフェース91及び通信バス94はパイプライン・ユニットの全てに共通している。單一外部バス・インターフェース91を含むことは、図4と連携されて先に議論されたようにパイプライン回路80（図4）上のリソースを解放する。パイプライン・ユニット78₁～78_nは全て單一回路ボード（図6には不図示）上に配置され得るか、各パイプライン・ユニットは各回路ボード上に配置され得るか、或は、グループ分けされた多数のパイプライン・ユニットが多数の回路ボード上にそれぞれ配置され得る。後者の2つの具現化例において、バス・インターフェース91は回路ボードの内の1つの上に配置される。代替的には、パイプライン・ユニット78₁～78_nは、その各々が図4と連携されて先に議論されたように各工業規格バス・インターフェース91を含み得て、よってその各々がパイプライン・バス50或はルータ61（図3）と直に通
40
50

信し得る。この具現化例において、パイプライン・ユニット $7_8, - 7_8$ は先に議論されたように單一或は多数の回路ボード上に配置され得る。

【0068】

パイプライン・ユニット $7_8, - 7_8$ の各々はホストプロセッサ42(図3)や相互のピアである。即ち、各パイプライン・ユニット 7_8 は通信バス94を介して他の任意のパイプライン・ユニットと直に通信でき、通信バス94、バス・インターフェース91、ルータ61(もしあれば)、並びに、パイプライン・バス50を介してホストプロセッサ42と通信できる。代替的には、パイプライン・ユニット $7_8, - 7_8$ の各々は各バス・インターフェース91を含み、各パイプライン・ユニットはルータ61(もしあれば)及びパイプライン・バス50を介してホストプロセッサ42と直に通信できる。10

【0069】

以下、多数のパイプライン・ユニット加速器44の動作は2つの例で説明される。

【0070】

第1例において、パイプライン・ユニット 7_8 はデータをパイプライン 7_8 に転送し、それがタイムクリティカル方式でデータを処理し、よって、パイプライン・ユニット $7_8, - 7_8$ は1つ或はそれ以上のSYNC信号を用いてデータ転送及び処理を同期する。典型的には、SYNC信号はタイムクリティカルな機能をトリガーするには充分に速いが、大きなハードウェア・リソースを必要とし、比較して、典型的には事象はタイムクリティカルな機能をトリガーするには充分に速くないが、著しくより少ないハードウェア・リソースを必要とする。先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献4で議論されたように、SYNC信号はピアからピアに直に経路指定されるので、例えばパイプライン・バス50(図3)や通信バス94をトラバースする事象よりもより迅速に機能をトリガーできる。しかし、それは個別に経路指定されるので、SYNC信号は、パイプライン回路80(図4)の経路指定ライン及びバッファ等の専用回路を必要とする。逆に、事象は既存のデータ転送下部構造(例えば、パイプライン・バス50及び通信バス94)を用いるので、より少ない専用ハードウェア・リソースを必要とする。結果として、設計者は全てであるが殆どのタイムクリティカル機能をトリガーするのに事象を使用しがちである。20

【0071】

第1として、パイプライン・ユニット 7_8 はデータを通信バス94に駆動することによってパイプライン・ユニット 7_8 に該データを送信する。典型的には、パイプライン・ユニット 7_8 はデータとパイプライン・ユニット 7_8 のアドレスを含むヘッダーとを含むメッセージを生成する。もしパイプライン・ユニット 7_8 がデータを多数のパイプライン・ユニット 7_8 に送信するつもりであれば、2つの方法の内の1つでそれを実行し得る。詳細には、パイプライン・ユニット 7_8 は個別メッセージを仕向先パイプライン・ユニット 7_8 の各々に順次送信し得て、各メッセージは各仕向先ユニットのアドレスを含むヘッダーを含む。代替的には、パイプライン・ユニット 7_8 は單一メッセージ・メモリ内にデータと各仕向先パイプライン・ユニットのアドレスを含むヘッダーとを含ませることによって、仕向先パイプライン・ユニット 7_8 の各々にデータを同時に送信し得る。データの送信は、先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献4に更に議論されている。3040

【0072】

次に、パイプライン・ユニット 7_8 はデータを受信する。パイプライン・ユニット $7_8, - 7_8$ は、各々、共通通信バス94と結合され、各パイプライン・ユニット $7_8, - 7_8$ はデータの意図された受取人であるか否かを決定する。例えば、各パイプライン・ユニット $7_8, - 7_8$ はそのアドレスがメッセージのヘッダー内に含まれているかを決定する。この例において、ユニット $7_8, - 7_8$ は、それらがデータの意図された受取人でないことを決定し、よってそのデータを無視。即ち、そのデータをそれらのデータ・メモリ92(図4)にロードしない。逆に、パイプライン・ユニット 7_8 はそれがデータの意図された受取人であることを決定し、よってそのデータをそのデータ・メモリ92に50

ロードする。データの受信は、先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献4に更に議論されている。

【0073】

次いで、パイプライン・ユニット78_oが受信データを処理する準備を為すと、パイプライン・ユニット78_o等のピア或は外部装置（不図示）がSYN C信号を脈動して、パイプライン・ユニット78_oにデータを適時に処理させる。SYN C信号を脈動するピア／装置が、パイプライン・ユニット78_oが受信データを処理する準備をいつ為したかを決定し得る多数の技術が存在する。例えば、ピア／装置は、パイプライン／ユニット78_oのデータ送信後、所定時間でSYN C信号を單に脈動し得る。おそらく、この所定時間は、パイプライン・ユニット78_oにデータを受信させてそれをそのデータ・メモリ92（図4）にロードさせるに充分な長さである。代替的にはパイプライン・ユニット78_oはSYN C信号を脈動して、ピア／装置に受信データを処理する準備が為されたことを知らせる。

10

【0074】

次に、脈動されたSYN C信号に応じて、パイプライン・ユニット78_oは受信データを処理する。パイプライン・ユニットによるデータの処理は、先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献4に更に議論されている。

【0075】

次いで、パイプライン・ユニット78_oがデータの処理を終了すると、ピア、外部装置（不図示）、或は、ユニット78_o自体は、SYN C信号を脈動し得て、パイプライン・ユニット78_oにより多くのデータを送信するよう通知する。

20

【0076】

第2例において、ホストプロセッサ42（図3）はデータをパイプライン78_oに転送し、それがそのデータを非タイムクリティカル方式で処理し、よってホストプロセッサ及びパイプライン・ユニット78_oは1つ或はそれ以上の事象を用いてデータ転送及び処理を先に議論された理由から同期する。

【0077】

第1として、ホストプロセッサ42（図3）はデータをパイプライン・バス50（図3）に駆動することによってパイプライン・ユニット78_oに該データを送信する。典型的には、ホストプロセッサ42はデータとパイプライン・ユニット78_oのアドレスを含むヘッダーとを含むメッセージを生成する。もしホストプロセッサ42がそのデータを多数のパイプライン・ユニット78_oに送信するつもりであれば、第1例と連携されて先に議論された2つの方法の内の1つでそれを実行し得る。

30

【0078】

次に、パイプライン・ユニット78_oはパイプライン・バス50（図3）から工業規格バス・インターフェース91及び通信バス94を介してデータを受信する。パイプライン・ユニット78_o—78_oが、各々、共通通信バス94と組合されているので、各パイプライン・ユニットはそれが第1例と連携して先に議論された方式でデータの意図された受取人であるかを決定する。

40

【0079】

次いで、パイプライン・ユニット78_oが受信データを処理する準備を為すと、ホストプロセッサ42（図3）等のピア或は外部装置（不図示）はパイプライン・バス50上か或は通信バス94の直上に事象を生成して、パイプライン・ユニット78_oにそのデータを適時に処理させる。事象を生成するピア／装置が、パイプライン・ユニット78_oが受信データを処理する準備をいつ為したかを決定し得る多数の技術が存在する。例えば、ピア／装置は、ホストプロセッサ42のデータの送信後、所定時間で事象を單に生成し得る。おそらく、この所定時間は、パイプライン・ユニット78_oにデータを受信させてそれをそのデータ・メモリ92（図4）にロードさせるに充分な長さである。代替的にはパイ

50

ブライン・ユニット78_oは事象を生成して、ピア／装置に受信データを処理する準備が為されたことを知らせる。

【0080】

次に、パイプライン・ユニット78_oは事象を受信する。事象の受信は、先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献4に更に議論されている。

【0081】

次いで、受信事象に応じて、パイプライン・ユニット78_oは受信データを処理する。パイプライン・ユニット78_oによるデータの処理は、先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献4に更に議論されている。
10

【0082】

次に、パイプライン・ユニット78_oがデータの処理を終了すると、ピア、外部装置（不図示）、或は、ユニット78_o自体は、事象を生成し得て、ホストプロセッサ42（図3）により多くのデータを送信するように通知する。

【0083】

図6で更に参照されるように、加速器44の代替的具現化例が想定されている。例えば、先に議論された動作の第1及び第2の例はSYNC信号及び事象をそれぞれ排他的に使用するが、加速器44はSYNC信号及び事象の双方を組み合わせて使用できることが想定されている。更には、他のピアは各データ・メモリ92内へのデータの半なるバルク記憶用に多数のパイプライン・ユニット78_o或は100の内の1つ或はそれ以上を使用できる。加えて、設計者はホストプロセッサ42（図3）を、ホストプロセッサの機能を実行する「ホスト」ピアと一緒に形成するパイプライン・ユニット78_o或は100の内の1つ或はそれ以上と置き換え可能である。更には、パイプライン・ユニット78_o或は100の1つ或はそれ以上は1つ或はそれ以上のメッセージ分配ピアとして作用し得る。例えば、ホストプロセッサ42が多数の加入者ピアへの伝送のためのメッセージを生成することを仮定する。ホストプロセッサ42はそのメッセージをメッセージ分配ピアに送信し、該ピアがそのメッセージを加入者ピアの各々に分配する。結果として、ホストプロセッサ42ではなくそのメッセージ分配ピアは、メッセージ分配の負荷を請け負い、よってホストプロセッサによってより多くの時間とリソースとを他のタスクに充てさせることを可能とする。
20
30

【0084】

図7は、本発明の別の実施例に従った多数のパイプライン・ユニット78_o（図4）或は100（図5）を有する加速器44（図3）のプロック線図である。

【0085】

図7の加速器44は、図7の加速器44が、パイプライン・ユニット78₁～78_nと、ホストプロセッサ42（図3）等の他のピアやパイプライン・バス50（図3）と結合されている装置（不図示）との間でのデータの経路指定に対する通信バス・ルータ110を含むことを除いて、図6の加速器44と同一である。 譲りの簡略化のため、図7の加速器44は多数のパイプライン・ユニット78₁～78_nを有するように議論されるが、その加速器は多数のパイプライン・ユニット100或はユニット78_o及び100の組み合わせを含み得ることを理解して頂きたい。
40

【0086】

通信バス・ルータ110は通信バス94の各分岐94₁～94_nを介してパイプライン・ユニット78₁～78_nと結合されると共に、バス112を介して工業規格バス・インターフェース91（もしあれば）と結合されている。代替的には、図6と連携して先に議論されたように、各パイプライン・ユニット78₁～78_nはボード上に各インターフェース91を含み得て、よって外部インターフェース91は省略ができるが、ルータ110は図3のパイプライン・バス50（或はもしあればルータ61）と直に結合されている。
50

【0087】

ルータ110は、パイプライン・バス50(図3)から各仕向先パイプライン・ユニット或は複数の同ユニット78₁～78_nまで信号を経路指定すると共に、ソース・パイプライン・ユニットから1つ或はそれ以上の仕向先パイプライン・ユニット若しくはパイプライン・バスまで信号を経路指定する。結果として、ルータ110はパイプライン・ユニット78₁～78_nの各々から通信バス94上の信号がそのパイプライン・ユニットに向けて意図されているかを決定する機能をオフロードする。このオフロード処理は各パイプライン・ユニット78₁～78_nのパイプライン回路80上のリソースを解放し得て、よって各パイプライン・ユニットの機能に関する増大を許容し得る。

【0088】

図7で更に参照されるように、ルータ110を伴う多数のパイプライン・ユニット加速器44の動作を以下に説明する。この動作は図6の加速器44に対しての先に記載されたものと類似しているので、以下の説明は図6及び図7の加速器間の動作差を強調している。

【0089】

第1例において、パイプライン・ユニット78₁はデータをパイプライン・ユニット78₂に転送し、それがそのデータをタイムクリティカル方式で処理し、よってパイプライン・ユニット78₁及び78₂は1つ或はそれ以上のSYNC信号を用いて、図6の第1例と連携して先に議論されたようにそのデータ転送及び処理を同期する。

【0090】

第1として、パイプライン・ユニット78₁はデータを通信バスの分岐94₁に駆動することによって該データをパイプライン・ユニット78₂に送信する。典型的には、パイプライン・ユニット78₁は、データとパイプライン・ユニット78₂のアドレスを含むヘッダーとを含むメッセージを生成する。

【0091】

次に、ルータ110はデータを受信し、そのデータの仕向先がパイプライン・ユニット78₂であることを決定し、そのデータを通信バスの分岐94₂に駆動する。典型的には、ルータ110は、データを含むメッセージのヘッダーを分析して、該ヘッダーから仕向先アドレスを抽出することによってデータの仕向先を決定する。結果として、ルータ110はデータの適切な仕向先を決定するので、パイプライン・ユニット78₂はルータからのデータをデータの意図された受取人であるかを決定することなしに単に受領できる。代替的には、パイプライン78₂はそれがデータの意図された受取人であるかを決定し得て、もしそれが意図された受取人でなければ例外を生成する(先行して引用された「改善された計算アーキテクチャ、関連システム、並びに、方法」と題された特許文献2、「改善された計算アーキテクチャを有する計算マシン、関連システム、並びに、方法」と題された特許文献3、「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献4で議論されたように)。パイプライン・ユニット78₂はこの例外をルータ110、工業規格バス・インターフェース91(もしあれば)、ルータ61(もしあれば)、並びに、パイプライン・バス50(図3)を介してホストプロセッサ42(図3)に送信できる。

【0092】

次いで、パイプライン・ユニット78₂はバス分岐94₂からデータをロードする。パイプライン・ユニットによるデータのローディングは、先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献4で更に議論されている。

【0093】

次に、パイプライン・ユニット78₂が受信データを処理する準備を為すと、パイプライン・ユニット78₂等のピア或は外部装置(不図示)がSYNC信号を振動して、図6の第1例と連携されて先に議論されたようにパイプライン・ユニット78₂にそのデータを適時に処理させる。

10

20

30

40

50

【0094】

次いで、脈動 S Y N C 信号に応じて、パイプライン・ユニット 78_aは図 6 の第 1 例と連携されて先に議論されたように受信データを処理する。

【0095】

次に、パイプライン・ユニット 78_aがデータの処理を終了すると、ピア、外部装置（不図示）、或は、ユニット 78_a自体は、S Y N C 信号を駆動し得て、パイプライン・ユニット 78_aにより多くのデータを送信するように通知する。

【0096】

第 2 例において、ホストプロセッサ 42（図 3）はデータをパイプライン 78_aに転送し、それがそのデータを非タイムクリティカル方式で処理し、よってホストプロセッサ及びパイプライン・ユニット 78_aは 1 つ或はそれ以上の事象を用いて、図 6 と連携して先に議論された理由のため、そのデータの転送及び処理を同期する。
10

【0097】

第 1 として、ホストプロセッサ 42（図 3）はデータをパイプライン・バス 50（図 3）に駆動することによってパイプライン・ユニット 78_aに送信する。典型的には、ホストプロセッサ 42 は、データとパイプライン・ユニット 78_aのアドレスを含むヘッダーとを含むメッセージを生成する。

【0098】

次に、ルータ 110 はそのデータを工業規格バス・インターフェース 91（もしあれば）及びバス 112 を介してパイプライン・バス 50（図 3）から受信する。
20

【0099】

次いで、ルータ 110 はそのデータの仕向先がパイプライン・ユニット 78_aであることを決定し、そのデータを通信バスの分岐 94_aに駆動する。典型的には、ルータ 110 は図 7 の第 1 例と連携して先に議論されたようにヘッダーの仕向先を決定する。結果として、ルータ 110 はデータの適切な仕向先を決定し、パイプライン・ユニット 78_aはルータからデータをデータの意図された受取人であることを決定することなしに単に受領することができる。代替的には、パイプライン 78_aはそれがデータの意図された受取人であるかを決定し得て、もしそれが意図された受取人でなければ例外を生成し（先行して引用された「改善された計算アーキテクチャ、関連システム、並びに、方法」と題された特許文献 2、「改善された計算アーキテクチャを有する計算マシン、関連システム、並びに、方法」と題された特許文献 3、「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献 4 で議論されたように）、図 6 の第 2 例と連携して先に議論されたように、その例外をホストプロセッサ 42（図 3）に送信する。
30

【0100】

次に、パイプライン・ユニット 78_aはバス分岐 94_aからデータをロードする。パイプライン・ユニットによるデータのローディングは、先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献 4 に更に議論されている。

【0101】

次いで、パイプライン・ユニット 78_aが受信データを処理する準備を為すと、ホストプロセッサ 42（図 3）等のピア或は外部装置（不図示）がパイプライン・バス 50 上或は通信バスの分岐 94_a、94_bの内の 1 つの上に事象を生成して、ユニット 78_aに図 6 の第 2 例と連携されて先に議論されたようにそのデータを適時に処理させる。
40

【0102】

次に、ルータ 110 は事象を受信して、それがパイプライン・ユニット 78_aに対して意図されていることを決定し、その事象をバス分岐 94_aに駆動する。

【0103】

次いで、パイプライン・ユニット 78_aはバス分岐 94_aから事象をロードする。パイプライン・ユニット 78_aによる事象のローディングは、先行して引用された「改善された計
50

算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献4に更に議論されている。

【0104】

次に、受信事象に応じて、パイプライン・ユニット78_oは受信データを処理する。

【0105】

次いで、パイプライン・ユニット78_oがデータの処理を終了すると、ピア、外部装置(不図示)、或は、ユニット78_o自体は、事象を生成し得て、ホストプロセッサ42(図3)により多くのデータを送信するように通知する。

【0106】

図7で更に参照されるように、動作の第1及び第2の例がそれぞれSYNC信号及び事象を併他的に使用するが、加速器44がSYNC信号及び事象の双方の組み合わせを使用可能であることが想定されている。 10

【0107】

図8は、本発明の実施例に従った多数のパイプライン・ユニット78(図4)或は100(図5)の多数グループ120を含む図3の加速器44のプロック線図である。パイプライン・ユニットの多数グループ120を含むことは加速器44の機能を増大し、関連動作を実行するパイプライン・ユニットをグループ分けすることによって設計者に加速器の効率を増大させることを可能としている。説明の簡略化のため、図8の加速器44は多数のパイプライン・ユニット78を有するように議論されるが、該加速器が多数のパイプライン・ユニット100或はユニット78及び100の組み合わせを含み得ることを理解して頂きたい。更には、パイプライン・ユニット78は工業規格バス・インターフェース91(このインターフェースはこの実施例においては外部にある)を含まないが、別の実施例では含んでもよい。 20

【0108】

加速器44はパイプライン・ユニット78の6つのグループ120₁～120₆を含み、各グループが3つのパイプライン・ユニットと、パイプライン・ユニットを相互に相互接続すると共に他のパイプライン・ユニット・グループと相互接続する各グループ内通信バス・ルータ110₁～110₆とを有する。加速器44が各々3つのパイプライン・ユニット78から成る6つのグループ120₁～120₆を含むように議論されているが、加速器の他の具現化例は任意数パイプライン・ユニットから成るグループの事実上の任意数を含み得て、グループの全てが同一数のパイプライン・ユニットを有する必要性はない。更には、通信バス・ルータ110₁～110₆は図6の加速器44と連携されて先に議論されたように省略され得る。 30

【0109】

パイプライン・ユニット・グループ120₁は3つのパイプライン・ユニット78₁～78₃を含み、それらが図7と連携されて先に議論されたものと類似した方式で通信バスの分岐94₁、94₂、並びに、94₃を介してグループ内通信バス・ルータ110₁と接続されている。他のグループ120₂～120₆は類似である。

【0110】

グループ120₁～120₃の通信バス・ルータ110₁～110₃は第1レベル・バス126₁の各分岐124₁～124₃を介して第1レベル・ルータ122₁と接続されている。ルータ122₁及びバス126₁はパイプライン・ユニット78₁～78₃が相互に通信することを可能としている。 40

【0111】

同様に、通信バス・ルータ110₄～110₆は第1レベル・バス126₂の各分岐124₄～124₆を介して第1レベル・ルータ122₂と接続されている。ルータ122₂及びバス126₂はパイプライン・ユニット78₄～78₆が相互に通信することを可能としている。

【0112】

第1レベル・ルータ122₁及び122₂は第2レベル・バス134の各分岐132₁～ 50

132₁を介して第2レベル・ルータ130と接続されている。ルータ130及びバス134はバイブライン・ユニット78₁～78₁₈が相互に通信すること及び以下に議論されるように他のピア／装置と通信することを可能としている。

【0113】

バイブライン・バス50及び第2バイブライン・バス136は各工業規格バス・インターフェース91₁及び91₂を介して第2レベル・ルータ130と結合されている。第2バイブライン・バス136はホストプロセッサ42(図3)等のピア、或は、バイブライン・バス50と結合されていないハードディスク・ドライブ(不図示)等の周辺機器と結合され得る。更には、バス50及び136の何れか或は双方はネットワーク或はインターネット(何れも不図示)を介してピア或は周辺機器と結合され得て、加速器44がホストプロセッサ42(図3)等の他のピアから遠隔的に位置決め可能である。
10

【0114】

バス138は1つ或はそれ以上のSYN C信号をバイブライン／ユニット78₁～78₁₈の全てと、ホストプロセッサ42(図3)等の他のピア或は装置(不図示)とに直に接続する。

【0115】

図8で更に参照されるように、動作の1つの例において、バイブライン・ユニット78から成る各グループ120₁～120₆は第2バイブライン・バス136と結合されたソナー・アレイ(不図示)の各センサからのデータを処理する。グループ120₁のバイブライン・ユニット78₁～78₃は單一ルータ110₁によって相互接続されているので、それらバイブライン・ユニットは、それらが他のグループ120₂～120₆のバイブライン・ユニット78₁～78₁₈に対してできるものより迅速に相互に通信することができる。このより高い通信速度は他のグループ120₂～120₆の各々においても存在する。結果として、設計者はデータを頻繁に転送するか成はざもなければそれらの間で通信するバイブライン・ユニットと一緒にグループ分けすることによって、加速器44の処理速度を増大することができる。
20

【0116】

一般に、バイブライン・ユニット78₁～78₁₈は相互に通信する共に、ホストプロセッサ42(図3)等のピア、及び、図7と連携されて先に議論されたものと類似の方式でバス50及び136と結合された装置と通信する。例えば、バス136と結合されたセンサ(不図示)は工業規格バス・インターフェース91₁、第2レベル・ルータ130、第1レベル・ルータ122₁、並びに、グループ内ルータ110₁を介してバイブライン・ユニット78₁と通信する。同様に、バイブライン・ユニット78₁は、ルータ110₁、122₁、110₃を介してバイブライン・ユニット78₁と通信すると共に、ルータ110₁、122₁、130、122₂、110₄を介してバイブライン・ユニット78₁₈と通信する。
30

【0117】

先行する議論は当業者が本発明を作製し使用することを可能とすべく提示されている。種々実施例への様々な変更は当業者には容易に明かであろうし、ここでの包括的な原則は本発明の精神及び範囲から逸脱することなしに他の実施例及び適用例に適用され得る。よって、本発明は図示された実施例に限定されることが意図されておらず、ここに開示された原理及び特徴と一致した最も広い範囲と一致されるべきものである。
40

【図面の簡単な説明】

【0118】

【図1】図1は、従来の多數プロセッサ・アーキテクチャを有する計算マシンのブロック線図である。

【図2】図2は、従来のハードウェアに組み込まれたバイブラインのブロック線図である。

【図3】図3は、本発明の実施例に従ったピア・ベクトル・アーキテクチャを有する計算マシンのブロック線図である。
50

【図 4】図 4 は、本発明の実施例に従った図 3 のパイプライン加速器におけるパイプライン・ユニットのブロック線図である。

【図 5】図 5 は、本発明の別の実施例に従った図 3 のパイプライン加速器におけるパイプライン・ユニットのブロック線図である。

【図 6】図 6 は、本発明の実施例に従った多数パイプライン・ユニットを含む図 3 のパイプライン加速器のブロック線図である。

【図 7】図 7 は、本発明の別の実施例に従った多数パイプライン・ユニットを含む図 3 のパイプライン加速器のブロック線図である。

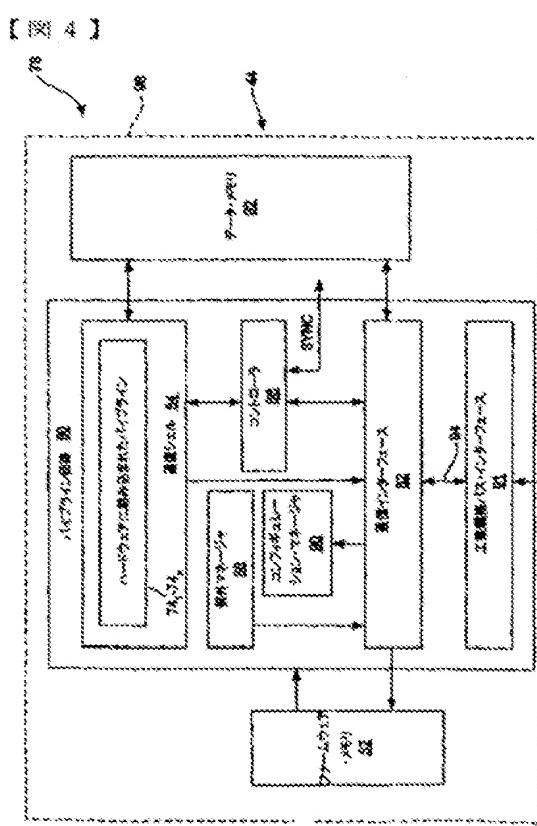
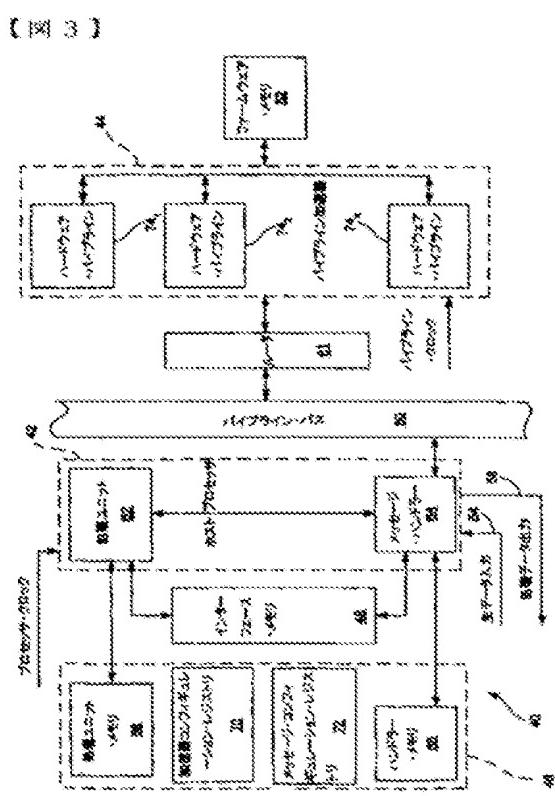
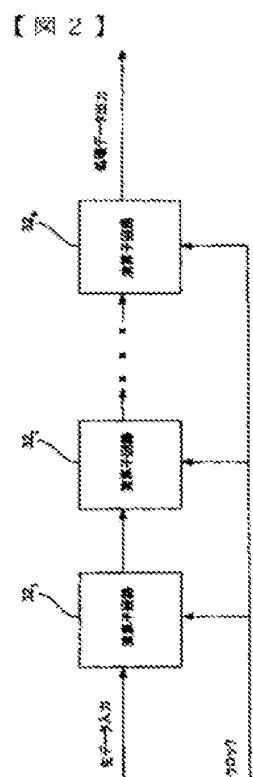
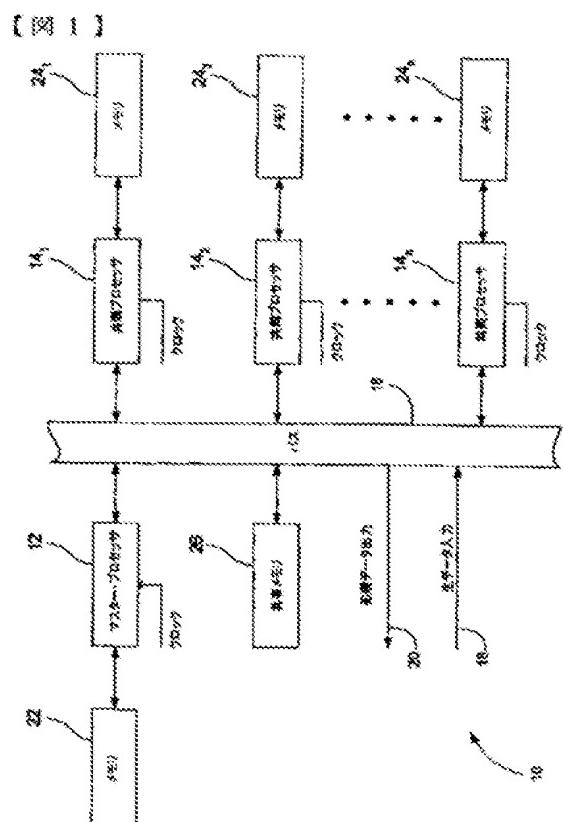
【図 8】図 8 は、本発明の実施例に従った多数パイプライン・ユニットからそれぞれが成る複数のループを含む図 3 のパイプライン加速器のブロック線図である。

10

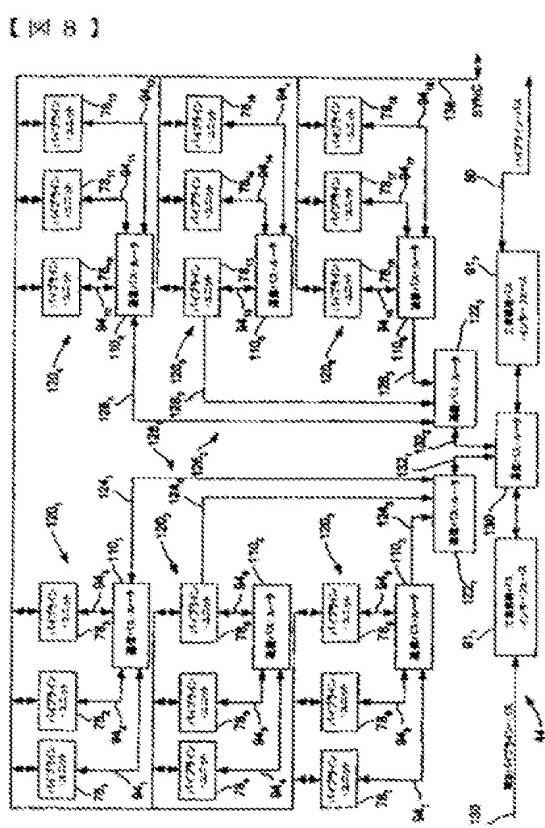
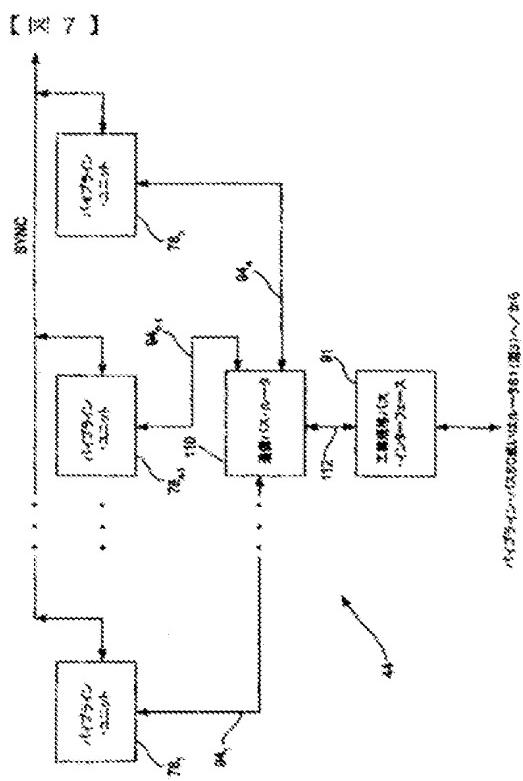
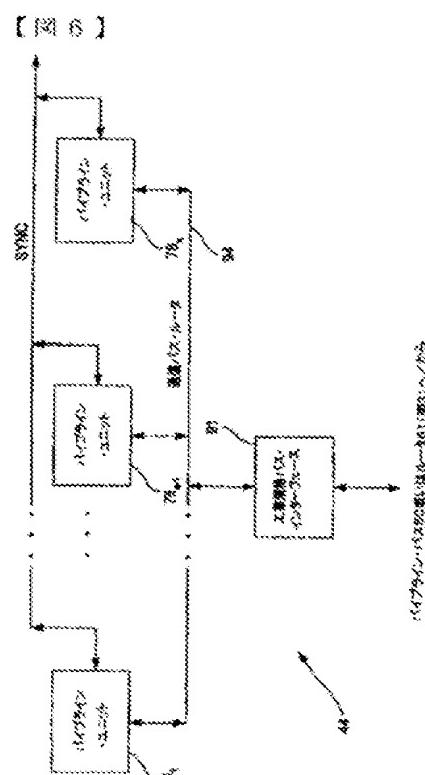
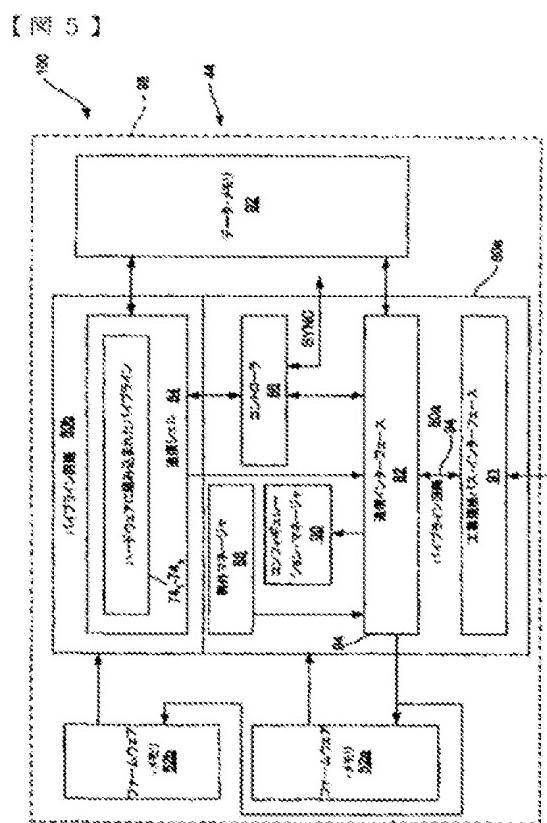
【符号の説明】

【0 1 1 9】

1 0	計算マシン	
1 4	共同プロセッサ	
4 0	ピア・ペクトル・マシン	
4 2	ホストプロセッサ	
4 4	パイプライン加速器	
4 6	プロセッサ・メモリ	
4 8	インターフェース・メモリ	
5 0	パイプライン・バス	20
5 2	ファームウェア・メモリ	
5 4	生データ入力ポート	
5 8	処理済みデータ出力ポート	
6 1	ルータ	
6 2	処理ユニット	
6 6	処理ユニット・メモリ	
6 8	ハンドラー・メモリ	
7 0	加速器コンフィギュレーション・レジストリ	
7 2	メッセージ・コンフィギュレーション・レジストリ	
7 4	ハードウェアに組み込まれたパイプライン	30
7 8	パイプライン・ユニット	
8 0	パイプライン回路	
8 6	パイプライン・コントローラ	
8 8	例外マネージャ	
9 0	コンフィギュレーション・マネージャ	
9 1	工業規格バス・インターフェース	
9 3	通信バス	



パトナシノバタクスル・リクルートホールディングス



フロントページの続き

- (31) 優先権主張番号 10/684,063
 (32) 優先日 平成15年10月9日(2003.10.9)
 (33) 優先権主張国 米国(US)
- (31) 優先権主張番号 10/684,057
 (32) 優先日 平成15年10月9日(2003.10.9)
 (33) 優先権主張国 米国(US)
- (31) 優先権主張番号 10/684,102
 (32) 優先日 平成15年10月9日(2003.10.9)
 (33) 優先権主張国 米国(US)
- (81) 指定国 AF(BW, GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, EG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), EP(AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OA(BF, BJ, CF, CG, CI, CR, GA, GN, GD, GW, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, GH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, CD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, ML, MN, MM, MX, MZ, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, GA, BG, US, CZ, VC, VN, YU, ZA, ZM, ZW)
- (74) 代理人 100135585
 弁理士 西尾 務
- (72) 発明者 シュルツ, ケニス, アール,
 アメリカ合衆国 バージニア州 201112 マナサッス, ゴールデン オータム コート 10
 506
- (72) 発明者 ラーブ, ジョン, ダブリュー,
 アメリカ合衆国 バージニア州 201110 マナサッス, リバー クレスト ロード 9350
- (72) 発明者 ジャクソン, ラリー
 アメリカ合衆国 バージニア州 201112 マナサッス, クレストロック ドライブ 130
 93
- (72) 発明者 ジョウンズ, マーク
 アメリカ合衆国 バージニア州 201200 セントレビル, オークマー ブレイス 15342
- (72) 発明者 カーサロ, トロイ
 アメリカ合衆国 バージニア州 22701 カルベバー, ケストラル コート 1524